ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ
INFORMATION TECHNOLOGY, COMPUTER SCIENCE AND MANAGEMENT







УДК 004.4
https://doi.org/10.23947/2687-1653-2023-23-1-76-84

Научная статья



Отказоустойчивый кластер хранилища данных для аналитических запросов в банковской сфере

В.В. Сивов ✉, **В.А. Богатырев**
Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Российская Федерация, г. Санкт-Петербург, Кронверкский пр., д. 49
✉ v.sivov777@gmail.com

Аннотация
Введение. Банковский сектор придает большое значение хранению данных, поскольку это критически важный аспект бизнес-операций. Объем данных в данной сфере неуклонно растет. С увеличением объемов данных, которые необходимо хранить, обрабатывать и анализировать, крайне важно выбрать подходящее решение для хранения данных и разработать необходимую архитектуру. Представленное исследование направлено на то, чтобы заполнить пробел в существующих знаниях СУБД, подходящих для банковского сектора, а также предложить способы для отказоустойчивого кластера хранения данных. Цель работы — анализ ключевых СУБД для аналитических запросов, определение приоритетов СУБД для банковского сектора и разработка отказоустойчивого кластера хранения данных. Для выполнения требований к производительности и масштабируемости предложено решение для хранения данных с отказоустойчивой архитектурой, отвечающее требованиям банковского сектора.
Материалы и методы. Анализ предметной области позволил создать набор характеристик, которым должна соответствовать СУБД для аналитических запросов (OLAP), выполнить сравнение некоторых популярных OLAP СУБД и предложить отказоустойчивую кластерную конфигурацию, написанную на языке xml, поддерживаемую СУБД ClickHouse. Автоматизация выполнена с помощью Ansible Playbooks. Он интегрирован с системой управления версиями Gitlab и шаблонами Jinja. Таким образом достигается быстрое развертывание конфигурации на всех нодах кластера.
Результаты исследования. Для баз данных OLAP были разработаны критерии, проведен сравнительный анализ нескольких популярных систем. В результате была предложена надежная кластерная конфигурация в банковской индустрии, которая удовлетворяет требованиям аналитических запросов. Для увеличения надежности и масштабируемости СУБД процесс развертывания был автоматизирован. Также приведены детальные схемы конфигурации кластера.
Обсуждение и заключения. Составленные критерии для OLAP СУБД позволяют определить необходимость данного решения в организации. Сравнение популярных СУБД может быть использовано организациями для минимизации затрат при выборе решения. Предлагаемая конфигурация кластера хранилища данных для аналитических запросов в банковской сфере позволит повысить надежность СУБД и удовлетворить требования к последующей масштабируемости. Автоматизация развертывания кластера путем механизма шаблонизации конфигурационных файлов в Ansible Playbooks позволяет настроить готовый кластер на новых серверах за минуты.

Ключевые слова: СУБД, OLAP, хранилище данных, ClickHouse, отказоустойчивый кластер.

Благодарности. Автор выражает благодарность В. А. Богатыреву, доктору технических наук, профессору кафедры вычислительной техники Университета ИТМО, почетному работнику науки и техники РФ, проводившему экспертные интервью совместно с автором статьи.

Для цитирования. Сивов В.В., Богатырев В.А. Отказоустойчивый кластер хранилища данных для аналитических запросов в банковской сфере. *Advanced Engineering Research (Rostov-on-Don).* 2023;23(1):76–84. https://doi.org/10.23947/2687-1653-2023-23-1-76-84

Original article

Data Warehouse Failover Cluster for Analytical Queries in Banking

Victor V Sivov ✉, **Vladimir A Bogatyrev**
ITMO University,49, Kronverksky Pr., St. Petersburg, Russian Federation
✉ v.sivov777@gmail.com

Abstract
Introduction. The banking sector assigns high priority to data storage, as it is a critical aspect of business operations. The volume of data in this area is steadily growing. With the increasing volume of data that needs to be stored, processed and analyzed, it is critically important to select a suitable data storage solution and develop the required architecture. The presented research is aimed at filling the gap in the existing knowledge of the data base management system (DBMS) suitable for the banking sector, as well as to suggest ways for a fault-tolerant data storage cluster. The purpose of the work is to analyze the key DBMS for analytical queries, determine the priorities of the DBMS for the banking sector, and develop a fault-tolerant data storage cluster. To meet the performance and scalability requirements, a data storage solution with a fault-tolerant architecture that meets the requirements of the banking sector has been proposed.
Materials and Methods. Domain analysis allowed us to create a set of characteristics that a DBMS for analytical queries (OnLine Analytical processing — OLAP) should correspond to, compare some popular DBMS OLAP, and offer a fault-tolerant cluster configuration written in xml, supported by the ClickHouse DBMS. Automation was done using Ansible Playbook. It was integrated with the Gitlab version control system and Jinja templates. Thus, rapid deployment of the configuration on all nodes of the cluster was achieved.
Results. For OLAP databases, criteria were developed and several popular systems were compared. As a result, a reliable cluster configuration that met the requirements of analytical queries has been proposed for the banking industry. To increase the reliability and scalability of the DBMS, the deployment process was automated. Detailed diagrams of the cluster configuration were also provided.
Discussion and Conclusions. The compiled criteria for the DBMS OLAP allowed us to determine the need for this solution in the organization. Comparison of popular DBMS can be used by organizations to minimize costs when selecting a solution. The proposed configuration of the data warehouse cluster for analytical queries in the banking sector will improve the reliability of the DBMS and meet the requirements for subsequent scalability. Automation of cluster deployment by the mechanism of templating configuration files in Ansible Playbook provides configuring a ready-made cluster on new servers in minutes.

Keywords: DBMS, OLAP, data warehouse, ClickHouse, failover cluster.

Acknowledgements. The author would like to thank V. A. Bogatyrev, Dr.Sci. (Engineering), professor of the Computer Engineering Department of ITMO University, Honorary Worker of Science and Technology of the Russian Federation, who conducted expert interviews together with the authors of the article.

For citation. Sivov V.V, BogatyrevV.A. Data Warehouse Failover Cluster for Analytical Queries in Banking. *Advanced Engineering Research (Rostov-on-Don).* 2023;23(1):76–84. https://doi.org/10.23947/2687-1653-2023-23-1-76-84

Введение. Хранилище данных в банковской сфере является одним из ключевых бизнес-факторов. Для обеспечения безопасности информации о клиентах и транзакциях, необходимо провести меры защиты, распределения и создания резервных копий. Для оперативного анализа сотрудники должны иметь возможность делать оперативные аналитические запросы в хранилище данных, при этом не мешая работе других процессов внутри организации и не вызывая большую нагрузку на само хранилище. Базы данных (Database) и хранилище данных (Data Warehouse) — это информационные системы, в которых производится хранение данных, но они используются и для решения различных задач. В статье описано, что делают такие системы, в чем основные различия между ними и почему их эффективное использование необходимо для развития бизнеса.

Многие организации допускают ошибки в проектировании архитектуры баз и хранилищ данных, упуская из виду аспекты информационной безопасности, масштабируемости и отказоустойчивости. Актуальность этой проблемы обусловлена интенсивным развитием систем в банках, расширением сфер их применения и увеличением количества данных, нуждающихся в постоянном анализе. Для оперативного анализа большого

количества данных необходимо хранилище, которое должно соответствовать всем требованиям надежности и безопасности.

Эффективные процессы принятия решений в бизнесе зависят от качественной информации. В современной конкурентной бизнес-среде требуется гибкий доступ к хранилищу данных, организованному таким образом, чтобы повысить производительность бизнеса, обеспечить быстрое, точное и актуальное понимание данных. Архитектура хранилища данных разработана для удовлетворения подобных требований и является основой этих процессов [1–5].

Цель работы — определение приоритетной СУБД для выполнения аналитических запросов в банковской сфере и проектирование отказоустойчивого кластера хранилища данных. Данное решение существенно повысит скорость выполнения аналитических запросов, решит проблемы с масштабируемостью и надежностью хранилища данных.

Материалы и методы. База данных (Database) хранит информацию в режиме реального времени об одной конкретной части бизнеса, ее основная задача заключается в обработке ежедневных транзакций. Базы данных используют оперативную обработку транзакций (OLTP) для быстрого удаления, вставки, замены и обновления большого количества коротких онлайн-транзакций.

Хранилище данных (Data warehouse) — это система, которая собирает данные из множества различных источников внутри организации для составления отчетов и анализа, используя оперативную аналитическую обработку (OLAP) для быстрого анализа больших объемов данных. Данная система сосредоточена на чтении, а не на изменении исторических данных из множества различных источников, поэтому соблюдение требований ACID (Atomic, Consistent, Isolated and Durable) менее строгое. Хранилища данных выполняют сложные функции агрегирования, анализа и сравнения данных для поддержки принятия управленческих решений в компаниях.

Хранилище в банковской сфере может содержать:

– учетную информацию о клиентах (персональные данные, адреса, телефоны);
– информацию о банковских продуктах и услугах (кредиты, депозиты, пластиковые карточки, мобильный банк и т. д.);
– данные об операциях (включая карточные) в минимальной детализации за последние 3 года;
– сведения о счетах, остатках на них и т.д.

Для удовлетворения потребностей в онлайн-аналитической обработке запросов (OLAP) существуют отдельные типы систем управления базами данных (СУБД) [3–6]. Каждая из систем имеет свои особенности в построении архитектуры.

Для проведения эффективного анализа соответствия указанным требованиям хранилища должны:

– обладать высокой вместимостью, способной вместить огромные объемы данных (миллиарды или триллионы строк);
– быть организованы в виде широких таблиц с множеством столбцов;
– выполнять запросы с небольшим количеством столбцов;
– иметь высокую скорость выполнения запросов (в миллисекундах или секундах);
– предусматривать большую часть запросов только на чтение;
– поддерживать быструю массовую загрузку данных при их обновлении (более 1000 строк за раз) и добавлении, но без их изменения;
– обладать высокой пропускной способностью для обработки одного запроса (до миллиардов строк);
– обладать высокой надежностью;
– обеспечивать безопасность и консистентность данных.

Для OLAP-сценария работы в банковской сфере предпочтительнее использовать колоночные аналитические СУБД, поскольку в них можно хранить много столбцов в таблице, что не будет сказываться на скорости чтения данных. Колоночные СУБД обеспечивают сильное сжатие данных в столбцах, так как в одной колонке таблицы данные, как правило, однотипные, чего не скажешь о строке. Они также позволяют на более маломощном оборудовании получить прирост скорости выполнения запросов в десятки раз. При этом, благодаря компрессии, данные будут занимать на диске в 5–10 раз меньше места, чем в случае с традиционными СУБД [7–11].

В ходе анализа требований выбраны следующие колоночные СУБД: ClickHouse, Vertica, Amazon Redshift.

ClickHouse является предпочтительным решением из-за следующих преимуществ: открытый исходный код; есть возможность определять некоторые или все структуры, которые будут храниться только в памяти; высокая скорость работы; хорошая степень сжатия данных; http и интерфейс командной строки; кластер можно

горизонтально масштабировать; высокая доступность; легкость установки и настройки. Установка осуществляется на серверах организации в изолированном сегменте, что отвечает требованиям безопасности для чувствительных данных в банковской сфере. Также СУБД включена в реестр отечественного программного обеспечения, поэтому позволяет внедрять данный программный продукт в государственные компании.

Решение Amazon Redshift предоставляется только в виде облачной службы. Для организаций из банковского сектора, которые не могут размещать свои данные в облаках по ряду причин, связанных с безопасностью, данный продукт теряет свою привлекательность.

Vertica является альтернативным вариантом ClickHouse с платной лицензией для крупных кластеров и возможностью установки на локальные серверы компании.

Ниже представлена реализация архитектуры распределенного хранилища данных. Для повышения отказоустойчивости и производительности предлагается реализация распределенного отказоустойчивого кластера ClickHouse с 3 шардами и 2 репликами.

Шардинг (горизонтальное масштабирование) позволяет записывать и хранить части данных в распределенном кластере, обрабатывать и считывать их параллельно на всех нодах кластера, увеличивая пропускную способность данных.

Репликация — копирование данных на несколько серверов, поэтому каждый бит данных можно найти на нескольких нодах.

Масштабируемость определяется шардированием или сегментацией данных. Надежность хранилища данных определяется репликацией данных [12–16].

Шардирование и репликация полностью независимы, за них отвечают разные процессы. Необходимо локализовать небольшие наборы данных на одном шарде и обеспечить достаточно ровное распределение по разным шардам в кластере. Для этого в качестве ключа шардирования рекомендуется брать значение хэш-функции из поля в таблице.

В зависимости от количества доступных ресурсов и серверов, предлагается реализовать эту конфигурацию на 3 или 6 нодах. Для производственной среды рекомендуется использовать кластер из 6 нод. Следует отметить, что репликация не зависит от механизмов шардирования и работает на уровне отдельных таблиц, а также, поскольку коэффициент репликации равен 2, то каждый шард представлен в 2 нодах [17–19]. Ниже описаны варианты конфигурации.

Схема логической топологии выглядит следующим образом:

3(Шард) × 2(Реплики) = Кластер Clickhouse из 6-ти нод.

Вероятность безотказной работы системы с 2 репликами и 3 шардами на 6 нодах равна:

$$P_\text{с} = [1 - (1 - p)^2]^3.$$

Вероятность безотказной работы — это объективная возможность того, что система без восстановлений проработает время t [7, 13].

Таким образом, таблица, содержащая 30 миллионов строк, будет распределена равномерно на 3 ноды кластера. Остальные 3 ноды будут хранить реплики данных. При выводе из строя одной из нод кластера, данные будут браться из другой доступной ноды, которая содержит ее реплику, тем самым достигается надежность [20]. Кластер из 6 нод изображен на рис. 1.

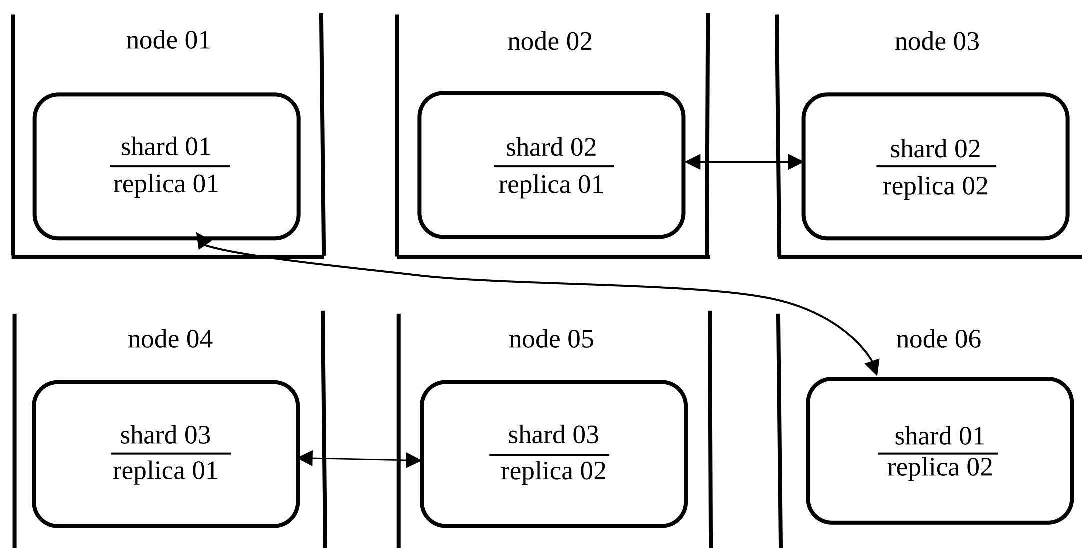


Рис. 1. Отказоустойчивый кластер из 6 нод

Для репликации данных и выполнения распределенных DDL запросов необходимо использовать +1 ноду с установленным ZooKeeper. Можно использовать также ClickHouse Keeper, совместимый с ZooKeeper, не требующий установки на отдельном сервере.

Пример фрагмента конфигурационного файла представлен на рис. 2, из которого видно, что шарде настроена репликация для 1-ой и 6-ой ноды.

```
<yandex>
    <remote_servers>
        <cluster_1>

            <shard>
                <weight>1</weight>
                <internal_replication>true</internal_replication>
                <replica>
                    <host>{{ node1 }}</host>
                    <port>9000</port>
                </replica>
                <replica>
                    <host>{{ node6 }}</host>
                    <port>9000</port>
                </replica>
            </shard>
```

Рис. 2. Фрагмент конфигурационного файла для 6 нод

Вариант конфигурации кластера из трех нод с цикличной репликацией изображен на рис. 3.

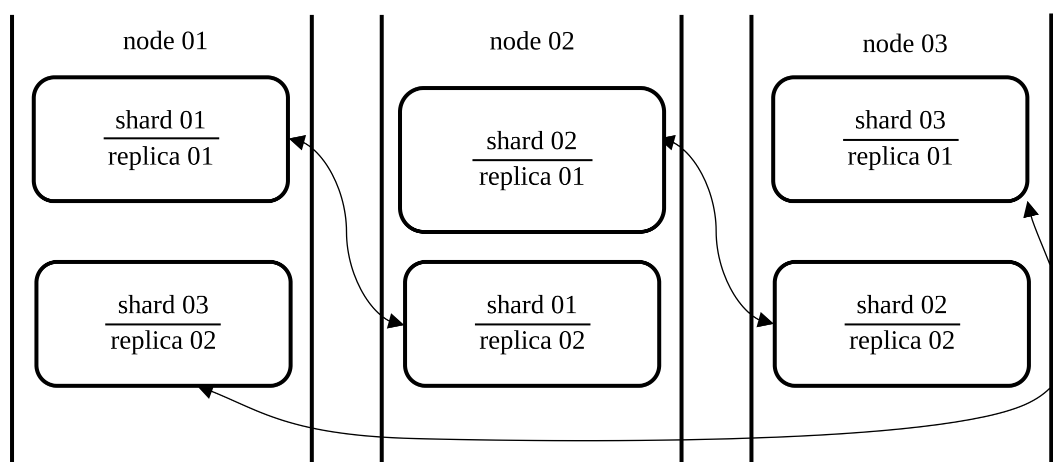


Рис. 3. Отказоустойчивый кластер из 3 нод

Для такой реализации требуется два разных сегмента, расположенных на каждой ноде. Основная проблема возникает из-за того, что каждый шард имеет одинаковое имя таблицы, ClickHouse не может отличить один шард/реплику от другого, когда они расположены на одном сервере.

Для решения данной проблемы необходимо:

– поместить каждый шард в отдельную базу данных (схему);
– установить default_database для каждого шарда;
– установить параметр internal_replication каждого шарда в true;
– использовать пустой параметр базы данных в сценарии DDL распределенной таблицы.

Для такой топологии в промышленной среде требуется 6 серверных нод, где каждый сервер хранит данные только одного сегмента, обходной путь для отдельной базы данных не требуется. Для экономии ресурсов в зоне разработки или тестирования можно использовать конфигурацию с 3 нодами.

Автоматизация выполнена с помощью Ansible Playbooks и интегрирована с системой управления версиями Gitlab. Таким образом достигается быстрое развертывание конфигурации на всех нодах кластера. При изменении конфигурации ее можно применить на всех нодах с помощью одной команды или развернуть новый кластер СУБД за несколько минут [21].

Результаты исследования. Отказоустойчивый кластер аналитической СУБД обеспечивает резервирование для важных компонентов системы, что позволяет продолжительно функционировать даже в случае возникновения ошибок в отдельных узлах кластера. Это делается за счет распределения нагрузки, репликации данных между узлами кластера и высокой надежности компонентов, используемых в кластере. Результатом является увеличение доступности и надежности аналитической СУБД, что важно для бизнеса, в котором аналитические запросы играют ключевую роль. Отказоустойчивая кластерная конфигурация хранилища данных для аналитических запросов в банковской сфере с учетом автоматизации процесса разворачивания позволяет повысить надежность аналитического хранилища данных и удовлетворить требования к масштабируемости. Разработанная задача по автоматизации развертывания кластера с использованием механизма шаблонизации конфигурационных файлов в Ansible Playbooks позволяет настроить готовый кластер на новых серверах за несколько минут. В задачи шаблона входят операции по установке необходимых пакетов, созданию необходимой конфигурации и запуску кластера.

Пример конфигурационных файлов для автоматического развертывания кластера СУБД приведен на рис. 4. Расширение j2 говорит о том, что они созданы с помощью механизма шаблонов Jinja. Специальные заполнители в шаблоне позволяют писать код, аналогичный синтаксису Python. В шаблон передаются параметры для автоматической вставки в финальный документ, тем самым достигается автоматическая сборка в зоны разработки, тестирования и промышленной эксплуатации, которая не требует ручного изменения файлов конфигурации.

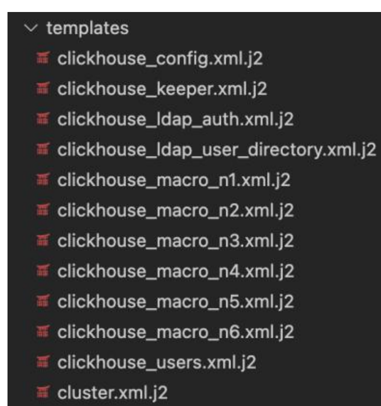


Рис. 4. Конфигурационные файлы

Описание конфигурационных файлов:

clickhouse_config.xml.j2 — общая конфигурация кластера;

clickhouse_keeper.xml.j2 — конфигурация zookeeper, которая отвечает за синхронизацию нод и репликацию;

clickhouse_ldap_auth.xml.j2 — конфигурация подключения к LDAP для обеспечения безопасности данных;

clickhouse_ldap_user_directory.xml.j2 — конфигурация с ролевой моделью по группам доступа для обеспечения безопасности данных;

clickhouse_macro_n1(6).xml.j2 — файлы макросов (для каждой ноды свой);

clickhouse_users.xml.j2 — конфигурационный файл создания локальных пользователей, необходимых для администрирования;

cluster.xml.j2 — конфигурационный файл кластера.

Для проверки надежности данной конфигурации проведен эксперимент, в ходе которого были загружены данные в кластер СУБД с коэффициентом репликации, равным 2. Созданы схемы dwh и таблицы cluster_test_data на каждой из нод кластера СУБД, а также создана распределенная таблица на кластере dwh.cluster_test_ data_ distributed. Строки таблицы dwh.test_ data_ distributed, распределенной по кластеру, равны 27 547 855. Ниже перечислены строки таблицы dwh.cluster_test_ data с каждой из нод кластера:

9186544 строк — 1-ая нода;

9182959 строк — 2-ая нода;

9182959 строк — 3-я нода;

9178352 строк — 4-ая нода;

9178352 строк — 5-ая нода;

9186544 строк — 6-ая нода.

Как можно заметить, таблица распределилась на весь кластер. Согласно конфигурации, приведенной на рис. 1, коэффициент репликации равен 2, значит, каждый блок данных будет представлен на 2 нодах. Это

можно увидеть из количества строк на нодах: шестая нода хранит копию первой, третья — копию второй, пятая — копию четвертой.

Отказоустойчивость данной конфигурации можно проверить попеременным выводом из строя нод в кластере. Для этого можно выключить ноду или остановить сервисы на одной из нод командой systemctl stop clickhouse-server. В ходе эксперимента была выполнена остановка сервисов СУБД на нодах кластера.

При одновременном отключении 3, 4, 6-ой или 1, 2, 5-ой нод, которые содержат реплики, пользователи продолжали получать данные из таблицы dwh.cluster_test_ data_ distributed, и количество строк было равным 27 547 855. При выводе из строя одной из нод, данные продолжали отображаться, а количество строк было равным 27 547 855. При одновременном отключении нод, которые содержат реплику и оригинальные данные, происходила потеря данных. Данную конфигурацию можно масштабировать на 12 нод, тогда коэффициент репликации будет равен 3, а коэффициент шардирования — 6.

Обсуждение и заключения. Предлагаемое решение может повысить скорость выполнения аналитических запросов, решить проблемы с масштабируемостью и надежностью хранилища данных в организациях банковской сферы. Авторами выполнена автоматизация развертывания кластера путем шаблонов в Ansible Playbooks, которая позволяет настроить готовый кластер на новых серверах за минуты. Данную конфигурацию можно масштабировать, увеличив количество нод и добавив их в конфигурационные файлы.

Обозначен набор характеристик, которым должна соответствовать OLAP СУБД, выполнено сравнение СУБД, предложена отказоустойчивая кластерная конфигурация хранилища данных для аналитических запросов в банковской сфере, выполнена автоматизация процесса разворачивания конфигурации. Подобное решение применимо для развертывания на FreeBSD, Linux, macOS. Приведены схемы конфигурации кластера. Данная конфигурация решит проблему, связанную с надежностью и масштабируемостью, которая часто встречается в организациях.

Список литературы

1. Sivov V.V. *Data Security in the Business Analytics System*. In: Proc. IV All-Russian Sci.-Pract. Conference with international participation “Information Systems and Technologies in Modeling and Control”. 2019. P. 142–145.

2. Solomon Negash, Paul Gray. Business Intelligence. In: *Handbook on Decision Support Systems 2.* Springer, Berlin, Heidelberg; 2008. P. 175–193.

3. Imhoff C., Galemmo N., Geiger J.G. *Mastering Data Warehouse Design: Relational and Dimensional Techniques.* John Wiley & Sons; 2003. 456 p.

4. Hugh J Watson. Tutorial: Business Intelligence – Past, Present, and Future. *Communications of the Association for Information Systems.* 2009;25:39. https://doi.org/10.17705/1CAIS.02539

5. Roscoe Hightower, Mohammad Shariat. Conceptualizing Business Intelligence Architecture. *Marketing Management Journal.* 2007;17:40–46.

6. Inmon W.H. *Building the Data Warehouse*, 4th ed. John Wiley & Sons; 2005. 576 p.

7. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Timely Redundant Service of Requests by a Sequence of Cluster. *CEUR Workshop Proceedings.* 2020;2590:1–12.

8. Henning Baars, Hans-George Kemper. Management Support with Structured and Unstructured Data — An Integrated Business Intelligence Framework. *Information Systems Management.* 2008;25:132–148.

9. Rachmiel A.G., Morgan N.P., Danielewski D. *Batch Management of Metadata in a Business Intelligence Architecture.* U.S. Patent No. 8,073,863 B2. 2011.

10. Dehne F., Eavis T., Rau-Chaplin A. The cgmCUBE Project: Optimizing Parallel Data Cube Generation for ROLAP. *Distributed and Parallel Databases.* 2006;19:29–62.

11. Bogatyrev V., Bogatyrev S., Bogatyrev A. Timely Redundant Service of Requests by a Sequence of Cluster. *CEUR Workshop Proceedings.* 2020;2590:1–12.

12. Milenin E.I., Sivov V.V. Simulation Model of Information Interaction of Measuring Devices in an Automated Environmental Monitoring System Based on IoT Technologies. *CEUR Workshop Proceedings.* 2021;2834:484–492.

13. Bogatyrev V.A., Bogatyrev S.V., Golubev I.Yu. Optimization and the Process of Task Distribution between Computer System Clusters. *Automatic Control and Computer Sciences.* 2012;46(3):103–111.

14. Cuzzocrea A., Il-Yeol Song, Davis K.C. *Analytics over Large-Scale Multidimensional Data: The Big Data Revolution!* In: Proc. DOLAP 2011, ACM 14th International Workshop on Data Warehousing and OLAP. 2011. P. 101–104. http://dx.doi.org/10.1145/2064676.2064695

15. Sivov V.V. *Sravnenie klyuchevykh programmnykh produktov dlya biznes-analitiki v bankovskoi sfere.* In: Proc. VI Int. Sci.-Pract. Conf. “Informatsionnye sistemy i tekhnologii v modelirovanii i upravlenii”. 2021. P. 281–287. (In Russ.)

16. Cuzzocrea A., Bertino E. Privacy Preserving OLAP over Distributed XML Data: A Theoretically-Sound Secure-Multiparty-Computation Approach. *Journal of Computer and System Sciences.* 2011;77:965–987. http://dx.doi.org/10.1016/j.jcss.2011.02.004

17. Cattell R. Scalable SQL and NoSQL Data Stores. *ACM SIGMOD Record.* 2010;12:12–27. https://doi.org/10.1145/1978915.1978919

18. Turban E., Sharda R., Delen D., et al. *Decision Support and Business Intelligence Systems* 9th ed. Pearson College Div; 2010. 696 p.

19. Olszak C.M., Ziemba E. Approach to Building and Implementing Business Intelligence Systems. *Interdisciplinary Journal of Information, Knowledge, and Management.* 2007;2:135–148. http://dx.doi.org/10.28945/105

20. Sarawagi S., Agrawal R., Megiddo N. *Discovery-Driven Exploration of OLAP Data Cubes*. In: Proc. Int. Conf. on Extending Database Technology – EDBT' 1998. Berlin: Springer, Berlin, Heidelberg; 1998. P. 168–182.

21. Anandarajan M., Anandarajan A., Srinivasan C.A. (eds.) *Business Intelligence Techniques. A Perspective from Accounting and Finance.* Berlin: Springer-Verlag Berlin; 2004. 268 p.

References

1. Sivov VV. *Data Security in the Business Analytics System*. In: Proc. IV All-Russian Sci.-Pract. Conference with international participation "Information Systems and Technologies in Modeling and Control". 2019. P. 142–145.

2. Solomon Negash, Paul Gray. Business Intelligence. In: *Handbook on Decision Support Systems 2*. Springer, Berlin, Heidelberg; 2008. P. 175–193.

3. Imhoff C, Galemmo N, Geiger JG. *Mastering Data Warehouse Design: Relational and Dimensional Techniques.* John Wiley & Sons; 2003. 456 p.

4. Hugh J Watson. Tutorial: Business Intelligence – Past, Present, and Future. *Communications of the Association for Information Systems*. 2009;25:39. https://doi.org/10.17705/1CAIS.02539

5. Roscoe Hightower, Mohammad Shariat. Conceptualizing Business Intelligence Architecture. *Marketing Management Journal.* 2007;17:40–46.

6. Inmon WH. Building the Data Warehouse, 4th ed. John Wiley & Sons; 2005. 576 p.

7. Bogatyrev VA, Bogatyrev SV, Bogatyrev AV. Timely Redundant Service of Requests by a Sequence of Cluster. *CEUR Workshop Proceedings.* 2020;2590:1–12.

8. Henning Baars, Hans-George Kemper. Management Support with Structured and Unstructured Data — An Integrated Business Intelligence Framework. *Information Systems Management*. 2008;25:132–148.

9. Rachmiel AG, Morgan NP, Danielewski D. *Batch Management of Metadata in a Business Intelligence Architecture.* U.S. Patent No. 8,073,863 B2. 2011.

10. Dehne F, Eavis T, Rau-Chaplin A. The cgmCUBE Project: Optimizing Parallel Data Cube Generation for ROLAP. *Distributed and Parallel Databases.* 2006;19:29–62.

11. Bogatyrev V, Bogatyrev S, Bogatyrev A. Timely Redundant Service of Requests by a Sequence of Cluster. *CEUR Workshop Proceedings.* 2020;2590:1–12.

12. Milenin EI, Sivov VV. Simulation Model of Information Interaction of Measuring Devices in an Automated Environmental Monitoring System Based on IoT Technologies. *CEUR Workshop Proceedings.* 2021;2834:484–492.

13. Bogatyrev VA, Bogatyrev SV, Golubev IYu. Optimization and the Process of Task Distribution between Computer System Clusters. *Automatic Control and Computer Sciences.* 2012;46(3):103–111.

14. Cuzzocrea A, Il-Yeol Song, Davis KC. *Analytics over Large-Scale Multidimensional Data: The Big Data Revolution!* In: Proc. DOLAP 2011, ACM 14th International Workshop on Data Warehousing and OLAP. 2011. P. 101–104. http://dx.doi.org/10.1145/2064676.2064695

15. Sivov VV. *Sravnenie klyuchevykh programmnykh produktov dlya biznes-analitiki v bankovskoi sfere.* In: Proc. VI Int. Sci.-Pract. Conf. "Informatsionnye sistemy i tekhnologii v modelirovanii i upravlenii". 2021. P. 281–287. (In Russ.)

16. Cuzzocrea A, Bertino E. Privacy Preserving OLAP over Distributed XML Data: A Theoretically-Sound Secure-Multiparty-Computation Approach. *Journal of Computer and System Sciences.* 2011;77:965–987. http://dx.doi.org/10.1016/j.jcss.2011.02.004

17. Cattell R. Scalable SQL and NoSQL Data Stores. *ACM SIGMOD Record.* 2010;12:12–27. https://doi.org/10.1145/1978915.1978919

18. Turban E, Sharda R, Delen D, et al. *Decision Support and Business Intelligence Systems* 9th ed. Pearson College Div; 2010. 696 p.

19. Olszak CM, Ziemba E. Approach to Building and Implementing Business Intelligence Systems. Interdisciplinary *Journal of Information, Knowledge, and Management.* 2007;2:135–148. http://dx.doi.org/10.28945/105

20. Sarawagi S, Agrawal R, Megiddo N. *Discovery-Driven Exploration of OLAP Data Cubes.* In: Proc. Int. Conf. on Extending Database Technology – EDBT' 1998. Berlin: Springer, Berlin, Heidelberg; 1998. P. 168–182.

21. Anandarajan M, Anandarajan A, Srinivasan CA. (eds.) *Business Intelligence Techniques. A Perspective from Accounting and Finance.* Berlin: Springer-Verlag Berlin; 2004. 268 p.

Об авторах:

Сивов Виктор Валерьевич, аспирант кафедры «Вычислительная техника» Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (197101, РФ, г. Санкт-Петербург, Кронверкский проспект, д. 49), ORCID, v.sivov777@gmail.com

Богатырев Владимир Анатольевич, доктор технических наук, профессор кафедры «Вычислительная техника» Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики (197101, РФ, г. Санкт-Петербург, Кронверкский проспект, д. 49), профессор кафедры «Информационная безопасность» Санкт-Петербургского государственного университета аэрокосмического приборостроения (190000, РФ, г. Санкт-Петербург, ул. Большая Морская, д. 67, лит. А), ORCID, ScopusID

Заявленный вклад соавторов:

В.В. Сивов — формирование основной концепции, цели и задачи исследования, проведение расчетов, подготовка текста, анализ результатов исследований, формирование выводов. В.А. Богатырев — научное руководство, анализ результатов исследований, доработка текста, корректировка выводов.

Поступила в редакцию 01.02.2023.
Поступила после рецензирования 17.02.2023.
Принята к публикации 20.02.2023.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the Authors:

Victor V Sivov, postgraduate of the Computer Science Department, ITMO University (49, Kronverksky Pr., St. Petersburg, 197101, RF), ORCID, v.sivov777@gmail.com

Vladimir A Bogatyrev, professor of the Computer Science Department, ITMO University (49, Kronverksky Pr., St. Petersburg, 197101, RF), professor of the Information Systems Security Department, State University of Aerospace Instrumentation (67, Bolshaya Morskaya St., Saint Petersburg, 190000, RF), Dr.Sci. (Eng.), ORCID, ScopusID

Claimed contributorship:

VV Sivov: basic concept formulation; research objectives and tasks; computational analysis; analysis of the research results; formulation of conclusions. VA Bogatyrev: academic advising; analysis of research results; revision of the text; correction of the conclusions.

Received 01.02.2023.
Revised 17.02.2023.
Accepted 20.02.2023.

Conflict of interest statement

The authors do not have any conflict of interest.

All authors have read and approved the final manuscript.